

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Digitální měřidlo hladiny kapaliny**

**Digital Liquid Level Meter**

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

## Zadání bakalářské práce

Student: **Vojtěch Bazgier**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Digitální měřidlo hladiny kapaliny**  
**Digital Liquid Level Meter**

### Zásady pro vypracování:

Cílem práce je konstrukce zařízení, které by umožňovalo měření množství kapalin v nádrži a následné zobrazení stavové informace zobrazovací jednotkou včetně možnosti jejího přenosu po sériovém rozhraní. Při konstrukci dbejte na dostatečnou přesnost měřidla. Výhodou by rovněž byla možnost připojení více měřících sond k jednomu centrálnímu modulu. Vytvořený modul bude sloužit zejména k měření neutrálních kapalin jako je voda ale také mírně agresivních (motorová nafta nebo oleje).

1. Nalezněte vhodný způsob měření hladiny kapaliny v nádrži s ohledem na zamýšlené použití.
2. Zkonstruuje hardwarový prototyp, který by zvolený způsob měření realizoval.
3. Implementujte firmware pro vzniklé hardwarové řešení jenž na základně znalosti rozměrů nádrže a výšky hladiny určil množství kapaliny.
4. Vzniklé měřidlo důkladně otestujte a stanovte jeho předpokládanou přesnost.

### Seznam doporučené odborné literatury:

MATOUŠEK, David. Práce s inteligentními displeji LCD . Praha : BEN - technická literatura, 2006. 224 s. ISBN 80-7300-121-7.

MATOUŠEK, David. Práce s mikrokontroléry Atmel AVR. Praha : BEN - technická literatura, 2006. 376 s. ISBN 80-7300-209-4.

BEJČEK, Ludvík; DĀDO, Stanislav; PLATIL, Antonín. Měření průtoku a výšky hladiny. Praha : BEN - technická literatura, 2006. 448 s. ISBN 80-7300-156-X.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Martin Milata**

Datum zadání: 18.11.2011

Datum odevzdání: 07.05.2013



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 28.4.2013

.....  
podpis studenta

## **Poděkování**

Rád bych poděkoval Ing. Martinu Milatovi za odbornou pomoc a konzultaci při vytváření této bakalářské práce.



## **Abstrakt**

Tato bakalářská práce se zabývá návrhem a realizací zařízení pro měření množství kapaliny v nádrži. Naměřený výsledek je zobrazován na zobrazovací jednotce a následně vysílán přes sériovou linku RS-232 do nadřazeného zařízení. Jádrem celé práce je osmi bitový mikroprocesor Atmel Mega8. Firmware, který celkovou funkčnost zařízení realizuje, je napsán v jazyce C.

## **Klíčová slova**

Atmel Mega8, RS-232, firmware

## **Abstract**

This thesis describes the design and realization of the device for measuring the quantity of liquid in the tank. The measured result is displayed on the display unit and then transmitted over a serial RS-232 link to the master device. The core of the thesis is an eight-bit microprocessor Atmel Mega8. Firmware that implements the complete functionality of the device, is written in C.

## **Key words**

Atmel Mega8, RS-232, firmware

## Seznam použitých zkratk a symbolů

**Tx** – Transmission - Vysílání

**Rx** – Receive - Příjem

**TTL** - Transistor-Transistor-Logic – Tranzistorově - Tranzistorová Logika

**EEPROM** - Electrically Erasable Programmable Read-Only Memory – Elektricky mazatelná i programovatelná paměť

**FLASH** - Elektricky mazatelná i programovatelná paměť

**USB** - Universal Serial Bus – Univerzální sériová sběrnice

**ISP** - In-System Programming – Programování mikroprocesoru bez vkládání do programátoru

**MOSI** - master output, slave input – Komunikace od Masterk Slave

**MISO** - master input, slave output - Komunikace od Slavek Master

**SCK** – Serial clock – Hodinový signál

**MASTER** – Řídící obvod

**SLAVE** – Podřízený obvod

**MIPS** - Million Instruction Per Second – Jednotka výkonnosti počítače

**Vcc** – Napájecí napětí

**GND** - Zem

**RISC** - Reduced Instruction Set Computing – Architektura mikroprocesorů

**USART** - Universal Synchronous / Asynchronous Receiver and Transmitter) – Zařízení pro sériovou komunikaci

**RS232** – Sériová linka

**BAUD** – Jednotka modulační rychlosti

**LCD** – Liquid Crystal Display – Zobrazovací jednotka

# Obsah

1	Úvod .....	1
2	Výběr metody měření hladiny kapaliny .....	2
2.1	Metoda měření pomocí ultrazvuku.....	2
2.1.1	Ultrazvuk .....	2
2.1.2	Měřicí modul Ultrasonic HC-SR04 .....	2
2.1.3	Specifikace měřicího modulu .....	3
2.1.4	Časový diagram a komunikace s modulem .....	4
3	Programátor mikroprocesoru.....	6
3.1	Podporované mikrokontroléry .....	6
3.2	Popis programátoru .....	7
3.3	Použitý software .....	8
3.4	Výběr mikroprocesoru.....	8
4	Atmel AVR ATmega8.....	9
4.1	Parametry mikroprocesoru .....	9
4.2	Vlastnosti a popis použitých modulů mikroprocesoru .....	10
4.2.1	Čítač/Časovač1 .....	10
4.2.2	USART (Univerzální Synchronní a Asynchronní sériová linka) .....	13
4.2.3	Popis USART .....	14
4.2.4	Popis registrů USART .....	16
4.2.5	Výpočet přenosové rychlosti USART .....	21
4.2.6	Komunikace po RS232.....	22
5	Převodník RS232.....	23
5.1	Praktické zapojení obvodu MAX232IN .....	23
6	LCD displej .....	25
6.1	Popis knihovny alank2 .....	26
6.1.1	Hlavičkový soubor hd44780.h.....	26

6.1.2	Hlavičkový soubor hd44780_settings.h .....	27
7	Návrh desky plošného spoje.....	28
7.1	Schéma zapojení.....	28
7.2	Popis napájecího obvodu.....	29
7.3	DPS.....	31
8	Firmware .....	32
8.1	Popis dílčích funkcí .....	32
8.2	Popis hlavního programu.....	35
9	Testování zařízení.....	37
9.1	Nastavení sériového přenosu .....	37
9.2	Inicializace měření .....	37
9.3	Přesnost zařízení.....	38
10	Závěr.....	40
11	Použitá literatura.....	41
12	Seznam obrázků .....	43
13	Seznam tabulek.....	44
14	Seznam příloh.....	45

---

# 1 Úvod

Cílem této bakalářské práce je konstrukce zařízení, které by umožňovalo měření množství neutrálních kapalin jako je voda, nebo také mírně agresivních látek, jako jsou oleje, či motorová nafta. V současné době je hlavním principem pro měření výšky hladiny kapalin používán bezkontaktní princip měření. Hlavní výhodou této metody je možnost měřit výšku hladiny i u látek, které jsou agresivní. V této bakalářské práci je použita metoda bezkontaktního měření a to pomocí ultrazvukového snímače. Princip celého zařízení spočívá v naměření výšky hladiny kapaliny obsažené v nádrži, a následně vypočítání celého objemu kapaliny nacházející se uvnitř této nádrže. Tvar nádrže a její rozměry jsou nastavovány do zařízení přes sériovou linku pomocí hyperterminálu v PC. Výsledek měření a vypočítaný objem kapaliny je následně zobrazen na zobrazovací jednotce a vyslán přes sériový port zpět do PC, kde může uživatel data dále zpracovávat.

Celkově se práce skládá ze tří základních částí. V první části se zabýváme nalezením vhodného způsobu pro určení výšky hladiny kapaliny a následně je popsán princip realizovaného způsobu. Následuje část, ve které jsou teoreticky popsány principy a fungování použitých hardwarových součástí, jako je mikroprocesor, obvod a pro sériovou komunikaci, LCD displej. Poslední, třetí část, se zabývá samotným návrhem zařízení. Jednak je popsáno celé schéma zapojení a deska plošného spoje, tak je i rozebrán a vysvětlen napsaný firmware a zařízení je otestováno. V závěru práce jsou prezentovány výsledky a možnost dalšího rozvoje tohoto zařízení v budoucnu.

---

## 2 Výběr metody měření hladiny kapaliny

Jedním z požadavků zadání bakalářské práce, bylo nalezení vhodného způsobu pro měření výšky kapaliny s ohledem na jeho použití. V současné době existuje mnoho způsobů a principů pro měření výšky kapalin. Jsou to například metody měření pomocí tlakoměru, metody založené na mechanického principu realizovány pomocí plováku, kapacitní metody nebo principy měření pomocí bezkontaktního čidla - ultrazvuku. Pro bakalářskou práci byl vybrán způsob měření pomocí ultrazvukové sondy, čili měření vzdálenosti použitím ultrazvukových vln. Mezi hlavní výhody ultrazvukových snímačů patří bezkontaktní měření, což je výhodné hlavně u agresivních látek nebo u vodní hladiny, která v zimě zamrzá.[1]

### 2.1 Metoda měření pomocí ultrazvuku

#### 2.1.1 Ultrazvuk

Ultrazvuk je akustické vlnění, neboli změna tlaku v prostředí, s frekvencí, která se pohybuje nad horní hranicí, která je slyšitelná pro člověka, tedy asi nad 20 kHz. Jednou ze základních fyzikálních vlastností ultrazvukových vln, na které je celý tento princip založen, je odraz. Odraz vzniká na rozhraní dvou prostředí, které mají rozdílnou hustotu. Čím je hustota jednoho prostředí větší od druhého prostředí, tím je odraz ultrazvuku silnější a nedochází k jeho absorpci [2]. Pro představu jsou v tabulce uvedeny hustoty látek, se kterými bychom se mohli setkat při měření:

Látka	Hustota [kg/m <sup>3</sup> ]
Voda	998,2
Vzduch	1,29
Motorová nafta	840
Olej transformátorový	866

Tabulka 1: Hustota vody a vzduchu o běžném tlaku a teplotě 20°C, zdroj: [3]

#### 2.1.2 Měřicí modul Ultrasonic HC-SR04

Pro měření množství hladiny byl vybrán ultrazvukový měřicí modul Ultrasonic HC-SR04. Tento modul se mimo jiné skládá z dvou hlavních částí, a to vysílače a přijímače ultrazvukových vln. Vysílací část (Tx) vysílá ultrazvukové vlny, ty se odráží od rozhraní dvou prostředí a odražená vlna je

---

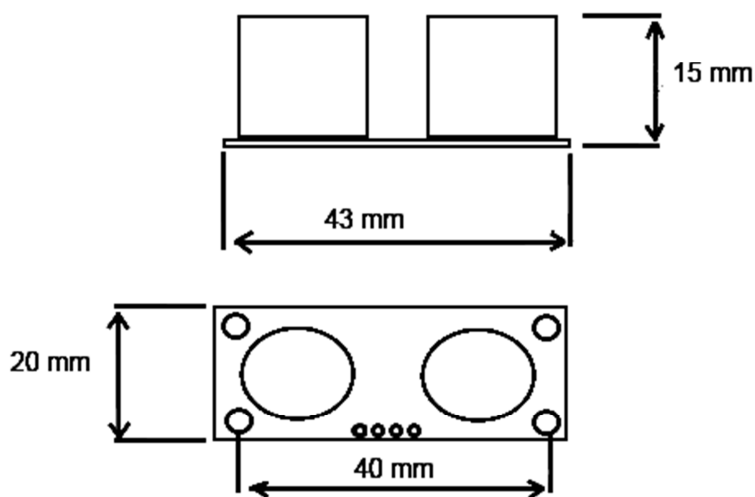
přijímaná přijímačem (Rx). Jelikož se jedná o ultrazvukové vlny, není žádným způsobem vyrušováno a ani omezováno okolí během měření.



Obrázek 1: Měřicí modul Ultrasonic HC-SR04, zdroj [4]

### 2.1.3 Specifikace měřicího modulu

V této kapitole si představíme specifikace měřicího modulu Ultrasonic HC-SR04. Jak už bylo uvedeno v předchozí kapitole, na modulu se nachází přijímací a vysílací část. Další nedílné součásti jsou 4 piny, sloužící pro napájení a komunikaci s modulem. Pin Vcc je určen pro přivedení napájecího napětí, které činí 5V. Dalším pinem je vstupní pin s názvem TRIG, sloužící pro spouštění měření. Na výstupním pinu ECHO přijímáme výsledek měření v podobě signálu, jehož délka je přímo úměrná měřené vzdálenosti. Posledním pinem je pin GND, k uzemnění modulu. Celé komunikace probíhá v TTL logice. Rozměry modulu můžeme vidět na obrázku 2.



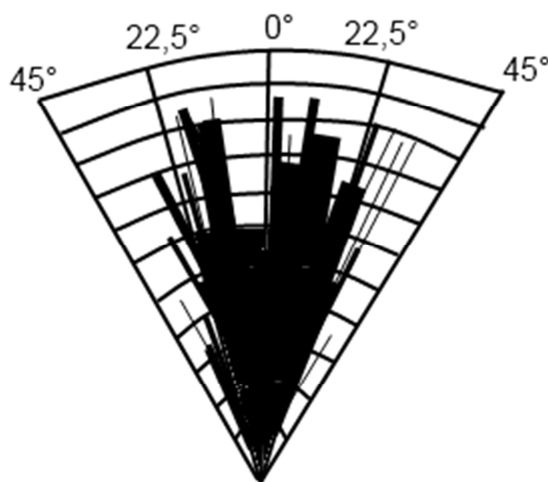
Obrázek 2: Rozměry modulu Ultrasonic HC-SR04, zdroj [4]

Parametry měřicího modulu Ultrasonic HC-SR04:

- Napájení modulu 5V, stejnosměrných



- Pracovní proud 15mA
- Efektivní úhel <15° viz *Obrázek 3*
- Použitelný měřicí rozsah vzdálenosti 2cm – 500 cm
- Chybovost 0,3 cm



*Obrázek 3: Efektivní uhel pro měření, zdroj [4]*

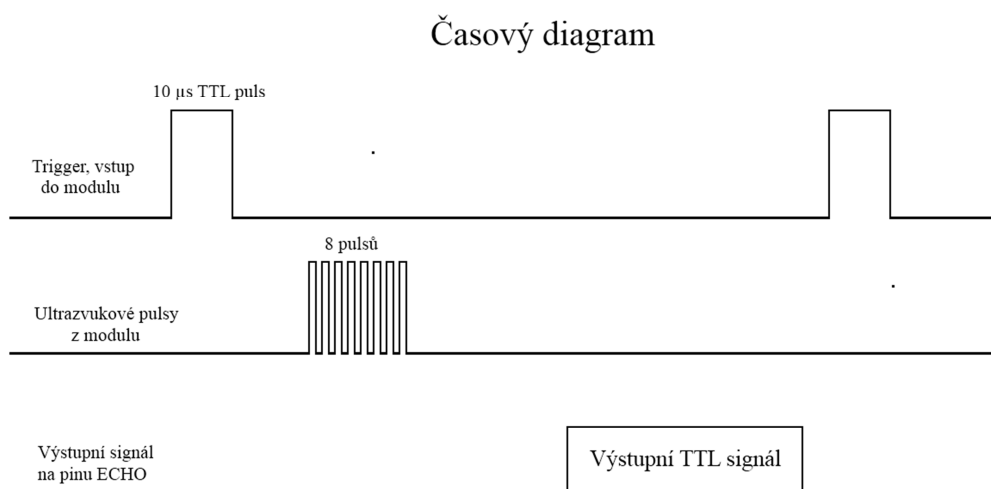
#### 2.1.4 Časový diagram a komunikace s modulem

Komunikace s modulem probíhá na dvou pinech (TRIG, ECHO). Jak můžeme vidět na časovém diagramu (*Obrázek 4*), měření vzdálenosti začne přivedením log. 1 (10  $\mu$ s TTL puls) na vstupní pin. Jakmile tento puls modul zaznamená, vyšle 8 pulzů o frekvenci 40 kHz, které se odrazí od překážky a vrátí se zpět. Na výstupním pinu ECHO je pak modulem generován TTL signál. Doba tohoto signálu na výstupním pinu je přímo úměrná vzdálenosti překážky od měřicího modulu. Trvání tohoto signálu se pohybuje přibližně od 150  $\mu$ s do 25 ms. Trvá-li puls po dobu 38 ms, nenachází se žádná překážka před modulem. Vypočet vzdálenosti překážky tedy lze definovat jako součin doby trvání ultrazvukových vln, s rychlosti ultrazvukových vln. Jelikož tento výpočet zahrnuje i dráhu, kdy se vlny vrací zpět, je třeba celý součin vydělit dvěma. Výsledný vzorec je uveden níže:

$$v = \frac{s * r}{2}$$

Kde:

- v ... vzdálenost překážky
- s ... šířka pulsu v mikrosekundách
- r ... rychlost ultrazvukových vln ve vzduch ( $r = 340 \text{ m/s} = 0,034 \text{ cm}/\mu\text{s}$ )[4]

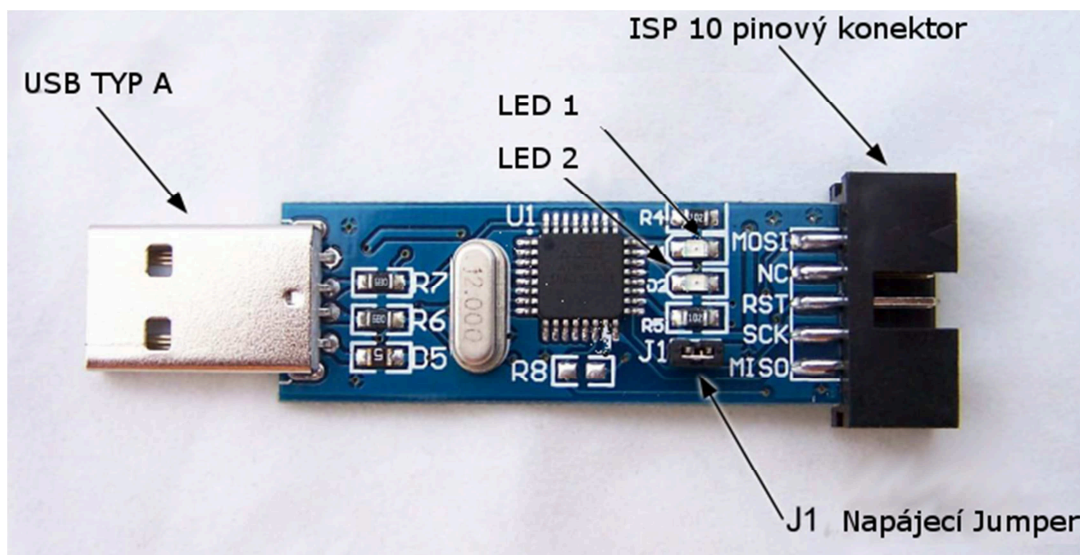


Obrázek 4: Časový diagram komunikace s modulem, zdroj [4]

---

### 3 Programátor mikroprocesoru

Jádrem hardwarového prototypu celé bakalářské práce je 8 bitový mikroprocesor AVR ATmega8. Mikroprocesor je programován i mazán pomocí elektrických impulzů. Je možno zapisovat a mazat program 10000 krát v případě flash paměti pro program a 100000 krát u paměti typu EEPROM, určené pro ukládání dat. K naprogramování mikroprocesoru ATmega8 bylo použito AC-PG-USBASP USBASP AVR programátoru. Jedná se o USB programátor, který je tvořen ATmega8 mikroprocesorem a dalšími pasivními komponenty. Je multiplatformní. To znamená, že je možno pracovat v operačních systémech Linux, Mac OS X i Windows [4].



Obrázek 5: Programátor, zdroj[5]

#### 3.1 Podporované mikrokontroléry

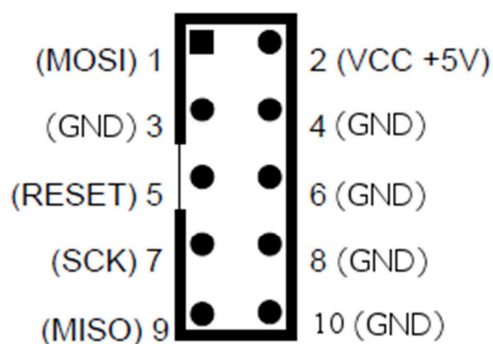
S výše zmiňovaným programátorem je možno programovat řadu mikrokontrolérů. V tabulce jsou uvedeny všechny podporované typy.

Podporované mikrokontroléry				
Série Mega				
ATmega8	ATmega48	ATmega88	ATmega168	ATmega328
ATmega103	ATmega128	ATmega1280	ATmega1281	ATmega16
ATmega161	ATmega162	ATmega163	ATmega164	ATmega169
ATmega2560	ATmega2561	ATmega32	ATmega324	ATmega329
ATmega3290	ATmega64	ATmega640	ATmega644	ATmega649
ATmega6490	ATmega8515	ATmega8535		
Série Tiny				
ATtiny12	ATtiny13	ATtiny15	ATtiny25	ATtiny26
ATtiny45	ATtiny85			
Série Classic				
AT90S1200	AT90S2313	AT90S2333	AT90S2343	AT90S4414
AT90S4433	AT90S4434	AT90S8515		
AT90S8535				
Série Can				
AT90CAN128				
Série PWN				
AT90PWM2	AT90PWM3			

Tabulka 2: Tabulka podporovaných mikrokontrolérů, zdroj[5]

### 3.2 Popis programátoru

Pro připojení programátoru k počítači slouží USB port typu A. Pro programování a propojení mikrokontroléru s programátorem slouží 10 pinový ISP konektor (Obrázek 6).



Obrázek 6: ISP 10 pinový konektor, zdroj[5]

Popis pinů ISP konektoru:

- MOSI - Master Out – Slave In
- SCK – Serial Clock

- 
- MISO - Master In – Slave Out
  - VCC – napájení 5 V
  - GND – zem
  - RESET – reset

Programátor a mikrokontrolér spolu komunikují třemi signály. Během programování je programátor v režimu MASTER a mikrokontrolér v režimu SLAVE. Master vysílá časovací signál (clock) na pinu SCK. S každým vyslaným pulsem na pinu SCK je vyslán bit na pinu MOSI z programátoru (master) na mikrokontrolér (slave) a ve stejný čas je poslán bit z mikrokontroléru (slave) do programátoru (master) na pinu MISO. Pinem reset programátor kontroluje reset mikrokontroléru.

Další součásti programátoru jsou dvě ledky. Led 1 nám signalizuje komunikaci s obvodem, který je programován. Led 2 indikuje, zda-li je programátor zapnutý.

Jumper J1 je určen k ovládání napájení na pinu Vcc. Je-li tento jumper aktivní, je na pinu Vcc napětí 5V a lze napájet zařízení, které chceme programovat. V opačném případě musím mít svůj vlastní zdroj napájení. Výstupní napětí lze měnit mezi 5V a 3,3V. Programátor obsahuje pojistku pro případ protékajícího proudu vyššího než 500 mA.

### 3.3 Použitý software

K vývoji a tvorbě firmwaru bylo použito prostředí WinAVR. Jedná se o kolekci volně šiřitelných nástrojů pro vývoj firmwaru pro Atmel AVR mikroprocesory na platformě Windows. Mezi nástroje patří například AVR-GCC kompilér, avrdude (programátor), avr-gdb (debugger) a další. Pro psaní samotného kódu sloužil nástroj Atmel Studio 6.0. Pro lepší přenositelnost mezi různými mikroprocesory a z důvodu vyhnutí se psaní zdlouhavého programu pomocí instrukcí v jazyce assembler, je celý firmware napsán ve vyšším programovacím jazyce C.

### 3.4 Výběr mikroprocesoru

Výběr mikroprocesoru byla další neodmyslitelná část bakalářské práce, která musela být realizována. Jelikož celý koncept práce nepočítá s velkou náročností na výpočetní výkon a není potřeba větších nátok na vstupní, výstupní porty, byl vybrán mikroprocesor AVR ATmega8

---

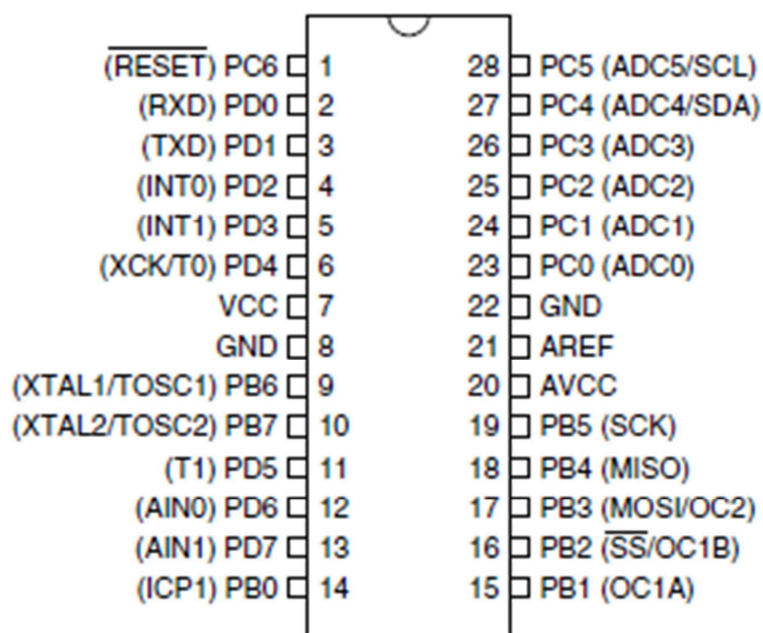
## 4 Atmel AVR ATmega8

Vybraný mikroprocesor Atmel AVR ATMEGA8L-8PU je 28 pinový, 8 bitový, nízko příkonový mikroprocesor. Je založen na Harwardské architektuře počítačů (odděluje paměť dat od paměti programu) a architektuře RISC, jejímž hlavním znakem je zjednodušení instrukční sady mikroprocesoru. Mezi jednu z mnoha vlastností tohoto mikroprocesoru patří 8 KB In-System programmable (ISP) Flash paměť, díky které je možno programovat mikroprocesor pomocí sériového ISP rozhraní přímo ve vyvíjené aplikaci.

### 4.1 Parametry mikroprocesoru

V následující kapitole si ukážeme některé z parametrů a vlastností mikroprocesoru Atmel AVR ATmega8. Podrobný popis mikroprocesoru je dostupný v katalogovém listu výrobce toho procesoru [6].

- 16MIPS při frekvenci 16MHz
- 8KB In-System Self-programovatelné Flash programové paměti
- 512B paměť EEPROM
- Dva 8 - bitové čítače/časovače s oddělenou děličkou, jeden porovnávací mód
- Jeden 16 - bitový čítač/časovač s oddělenou děličkou, porovnávací mód, a zachytávací mód
- Čítač čítající ve skutečném čase s odděleným oscilátorem
- Tři PWM kanály
- Šesti kanálový AD převodník s 10 bitovou přesností
- Programovatelný Sériový USART
- Programovatelný Watchdog časovač
- Vnitřní kalibrovaný oscilátor
- Vnější i vnitřní zdroje přerušení
- Operační napětí - 4,5 V až 5,5 V
- Frekvence čipu – 0 MHz až 16 MHz
- 23 vstupně, výstupních linek



Obrázek 7: Rozložení pinů mikroprocesorů ATmega8, zdroj[6]

Dále mikroprocesor obsahuje interní oscilátor, který je použit při realizaci této bakalářské práce. Jeho defaultní frekvence je 1 MHz. Na tuto frekvenci je mikroprocesor nastaven a kalibrován při každém jeho startu. Celý realizovaný obvod bude tedy na této frekvenci vyvíjen.

## 4.2 Vlastnosti a popis použitých modulů mikroprocesoru

Cílem této kapitoly je popis modulů mikroprocesoru ATmega8, které jsou použity při vývoji a realizaci této bakalářské práce. Tyto moduly patří k důležitým částem celé práce a vybavení mikroprocesoru ATmega8, bez nichž by se celá práce značně zkomplikovala. Popis dalších modulů, které nejsou v této práci použity, jsou dostupné v katalogovém listu výrobce [6].

### 4.2.1 Čítač/Časovač1

Jedná se o šestnácti bitový modul mikroprocesoru. To znamená, že počítadlo může nabývat až 65536 hodnot. Umožňuje přesné časování a spouštění programu, generování signálu i jeho časování při měření. Mezi jeho základní vlastnosti patří:

- 16 bitový
- Dvě nezávislé výstupní porovnávací jednotky
- Jedna jednotka umožňující zachytávání aktuální hodnoty čítače/časovače1 na událost Input Capture (Input Capture Unit)

- 
- The diagram illustrates the internal architecture of the TCCrN module. It features a central 'Ovládací logika' (Control Logic) block that manages the module's operation. Key components include:
- Čítač/časovač TCNTn**: A counter/timer block that receives 'hodnota' (value) and 'směr' (direction) signals and outputs to a series of comparators.
  - OCnA and OCnB**: Output Compare registers that generate 'OCFnA přerušení' and 'OCFnB přerušení' (interrupts) when the counter matches their values.
  - ICFn přerušení**: Input Capture interrupt generated when an input value matches the counter.
  - Detektor hrany**: Edge detector blocks that identify signal transitions.
  - Generace průběhu**: Waveform generation blocks that produce 'OCFnA přerušení' and 'OCFnB přerušení' based on the counter's state.
  - Omezovač šumu**: A noise filter block that processes the output of the edge detector.
  - TCCrNA and TCCrNB**: Output registers that store the final processed values.
- The module is connected to a 'DATOVÁ SBĚRNICE' (Data Bus) for data exchange. The control logic also manages the 'TOP' and 'BOTTOM' signals, which are used for signal selection and comparison.

Čítač/časovač1 obsahuje 16 bitový registr TCNT1, do kterého se ukládá aktuální obsah čítače. Inkrementace obsahu registru TCNT1 je prováděná několika různými způsoby. Prvním způsobem je inkrementace přes externí vstup T1, kdy je obsah registru navýšen při každé zachycené změně (hraně) vstupního signálu. Obsah registru lze zvyšovat i vnitřně, buď přímo nebo přes předděličku, v závislosti na hodinovém signálu.

---

11



Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
čtení/zápis	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Počáteční hod.	0	0	0	0	0	0	0	0	

Obrázek 9: Registr TCCR1B, zdroj[6]

Popis registru TCCR1B:

- **ICNC1 (Input capture noise canceller)** – Je-li v log. 1 je aktivován omezovač šumu na pinu ICP1
- **ICES1 (Input Capture Edge Select)** – Je-li v log. 1 provádí se zachytávání aktuálního obsahu registru TCNT1 do registru ICR1 na náběžnou hranu vstupního signálu na pinu ICP1. Při nastavení do log. 0 se provádí zachytávání na sestupnou hranu
- **WGM13, WGM12** – Nastavení PWM módu
- **CS12, CS11, CS10** – Výběr zdroje hodin a předděličky. Zdroj hodin můžeme nastavit jako vnitřní zdroj (přímý nebo zmenšen, je-li přiveden přes předděličku) nebo vnější zdroj na pinu T1.

CS12	CS11	CS10	Popis
0	0	0	Žádný zdroj hod. signálu (čítač/časovač zastaven)
0	0	1	Clk/1 (bez předděličky)
0	1	0	Clk/8 (Z předděličky)
0	1	1	Clk/64 (Z předděličky)
1	0	0	Clk/256 (Z předděličky)
1	0	1	Clk/1024 (Z předděličky)
1	1	0	Externí zdroj hodin na pinu T1. (Sestupná hrana)
1	1	1	Externí zdroj hodin na pinu T1. (Vzestupná hrana)

Tabulka 3: Nastavení zdroje hodin a předděličky, zdroj[6]

Pro nastavení a povolení jednotlivých přerušení od čítače/časovače1 slouží registr TIMSK (Timer/counter Interrupt Mask Register).

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	–	TOIE0	TIMSK
čtení/zápis	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Počáteční hod.	0	0	0	0	0	0	0	0	

Obrázek 10: Registr TIMSK, zdroj[6]

Popis registru TIMSK:

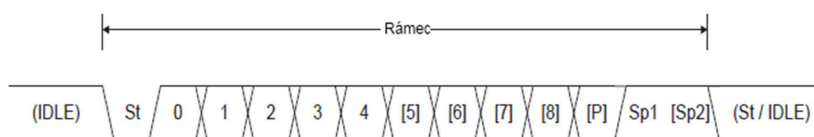
- **TICIE1(Timer/counter 1 Input Capture Interrupt Enable)** – Povolení přerušení při příchodu signálu na pin ICP1
- **OCIE1A(Output compare A Interrupt Enable)** – Povolení přerušení, rovná-li se obsah registru TCNT1 s registrem OCR1A
- **OCIE1B (Output compare B Interrupt Enable)** - Povolení přerušení, rovná-li se obsah registru TCNT1 s registrem OCR1B
- **TOIE1 (Timer/counter 1 Overflow Interrupt Enable)** – Povolení přerušení při přetečení registru TCNT1

#### 4.2.2 USART (Univerzální Synchronní a Asynchronní sériová linka)

Sériový přenos dat lze rozdělit na synchronní a asynchronní komunikaci. Mezi jejich hlavní rozdíly patří to, že v případě synchronního přenosu potřebujeme hodinový signál pro synchronizaci přenosu dat. Synchronní přenos dosahuje vyšších rychlostí přenosu dat. Avšak asynchronní komunikace se používá ve více případech, a to i z důvodu, že počítač s rozhraním RS-232 dokáže komunikovat pouze asynchronně.

Datový rámec v případě asynchronního přenosu dat tvoří následující bity:

- Start bit – inicializuje přenos, ze synchronizuje vysílač i přijímač, je vždy v log. 0
- Datové bit – 5 až 9 bitů
- Paritní bit – kontrola dat, může být lichá nebo sudá
- Stop bit – ukončuje přenos, vždy v log. 1
- Nečinný stav – neprobíhá-li přenos je datová linka v log. 1



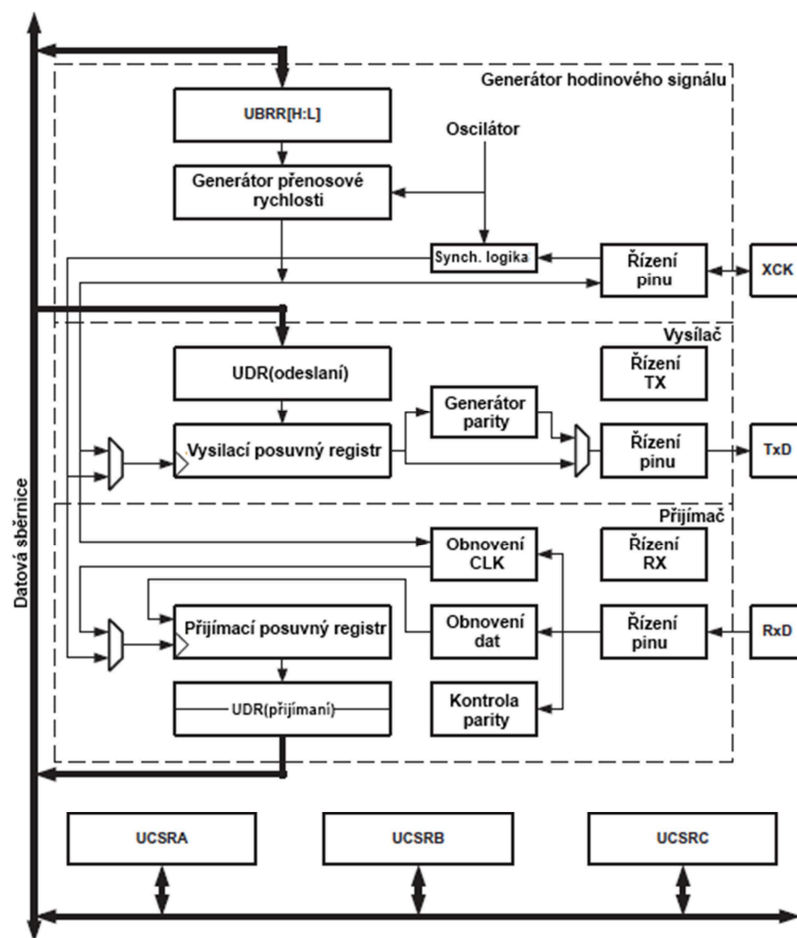
- St** Start bit, vždy v logické nule
- (n)** Datové bity (0 až 8)
- P** Paritní bit (sudá nebo lichá parita)
- Sp** Stop bit, vždy v logické jedna
- IDLE** Žádný přenos na komunikační lince (RxD nebo TxD) IDLE musí být v log. 1

*Obrázek 11: Datový rámec asynchronního přenosu, zdroj[6]*

#### 4.2.3 Popis USART

V této části si popíšeme hardwarovou součást mikroprocesoru ATmega8, která slouží pro obousměrný přenos dat přes sériovou linku. Jedná se o USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter). Mezi hlavní výhody modulu patří:

- Full Duplex operace (Nezávislé přijímací i vysílací registry)
- Asynchronní nebo Synchronní operace
- Podpora rámců s 5,6,7,8 nebo 9 data bity a 1 nebo 2 stop bity
- Kontrola paritou
- Detekce chyby rámce
- Omezovač šumu
- Tři oddělené zdroje přerušení (dokončení vysílání, vyprázdnění vysílacího registru, dokončení příjmu dat)
- Zdvojnásobená rychlost přenosu dat při Asynchronní komunikaci



Obrázek 12: Blokové schéma USART, zdroj[6]

Modul USART se skládá ze tří základních modulů a řídicích registrů:

- **generátor hodinového signálu** - Generátor hodinového signálu se skládá ze synchronizačních obvodů, sloužící pro řízení obvodu, je-li přiveden externí zdroj hodinového signálu v případě synchronního slave režimu. XCK pin je použit pouze v případě synchronního přenosového módu.
- **Vysílač** - Vysílač tvoří zapisovací buffer, posuvný registr, generátor parity, a řídicí obvod pro přenos různých formátů rámců
- **Přijímač** - Přijímač je díky tomu, že obsahuje modul pro generování hodinového signálu a jednotku pro obnovu dat, nejsložitější částí celé jednotky USART. Jednotky pro obnovu dat, které rozpoznávají, v jaké logické hodnotě se přijatý bit nachází, se používají pouze v případě asynchronního příjmu dat. Kromě této jednotky, přijímač obsahuje modul kontroly parity, ovládací logiku, posuvné registry a dvouúrovňový

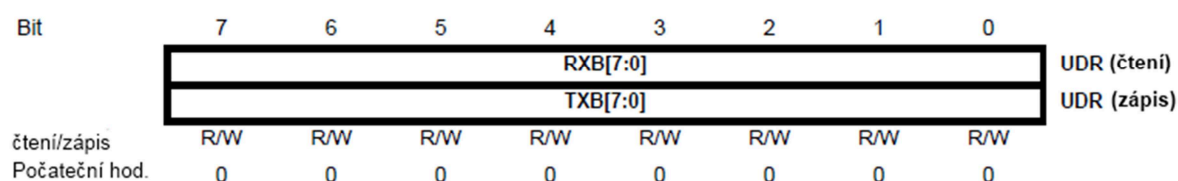
přijímací registr UDR. Podporuje taktéž stejné formáty rámců jako vysílač a je uzpůsoben pro detekci chyb, přetečení dat a chybu parity.

- **řídící registry** - Řídící registry jsou společné pro všechny části USART. Mezi registry patří jeden 16 -ti bitový registr UBRR(pro nastavení přenosové rychlosti), tři 8 bitové řídicí a stavové registry UCSRA, UCSRB, UCSRB a jeden 8 bitový registr UDR pro ukládání a zápis dat.

#### 4.2.4 Popis registrů USART

- Registr UDR (USART I/O Data Register)

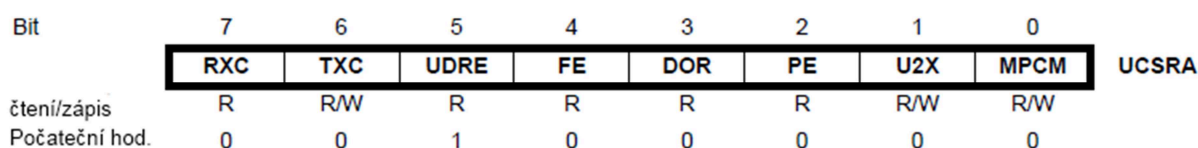
Jedná se o osmi bitový registr, do něhož se ukládají data přijata přes UART a taktéž data, která se budou přes UART vysílat.



Obrázek 13: Registr UDR, zdroj[6]

- Registr USCRA (Control and Status Register A)

Osmi bitový řídicí a stavový registr, určen pro nastavování USART a zjišťování stavu USART.



Obrázek 14: Registr USCRA, zdroj[6]

Popis jednotlivých bitů v registru USCRA:

- **Bit 7 – RXC:** USART Receive Complete – Tento bit je nastaven na log. 1, jsou-li nepřečtená data obsažena v přijímacím bufferu. Je-li buffer prázdný, bit je nastaven na log. 0. Pokud je přijímač zakázán, buffer je vyprázdněn a bit je nastaven na log. 0. RXC příznak lze použít ke generování přerušení.
- **Bit 6 – TXC:** USART Transmit Complete – Bit TXC je nastaven na log. 1, je-li celý datový rámec vyslán z vysílacího posuvného registru a žádná další nová data nejsou

---

obsažena ve vysílacím bufferu (UDR). TXC příznakový bit je automaticky vymazán, je-li vysílání dokončeno a vykoná-li se příslušné přerušení nebo je vymazán zapsáním log. 1 do příslušné lokace. Tento příznakový bit může generovat příslušné přerušení.

- **Bit 5 – UDRE:** USART Data Register Empty – Příznakový bit UDRE indikuje, jestli vysílací buffer (UDR) je připraven přijímat nová data. Jestliže je UDRE v log. 1 je buffer prázdný a připraven na zápis. V log. 1 je nastaven po resetu, aby indikoval, že je vysílač připraven. Příznakový bit UDRE může generovat příslušné přerušení.
- **Bit 4 – FE:** Frame Error – Tento bit je nastaven na log. 1, má-li znak v přijímacím registru chybu rámce (nastává v případě, že první stop bit v přijímacím bufferu je log. 0) Příznakový bit FE je platný, dokud je čten obsah z přijímacího bufferu (UDR). FE bit je nastaven na log. 0, je-li stop bit přijatých dat v log. 1. Při zápisu do registru UCSRA se tento příznakový bit nastaví do log. 0.
- **Bit 3 – DOR:** Data OverRun – Dojde-li přetečení dat je nastaven tento bit do log. 1. Přetečení nastává v případě že: buffer je plný (dva znaky), nový znak čeká v přijímacím posuvném registru a je detekován nový start bit. Tento bit je platný, dokud je čteno z přijímacího bufferu (UDR). Při zápisu do registru UCSRA se tento příznakový bit nastaví do log. 0.
- **Bit 2 – PE:** Parity Error – Příznakový bit PE se nastaví do log. 1, jestliže znak v přijímacím bufferu obsahuje chybu parity a kontrola parity byla povolena (UPM = 1). Tento bit je platný, dokud je čten z přijímacího bufferu (UDR). Při zápisu do registru UCSRA se tento příznakový bit nastaví do log. 0.
- **Bit 1 – U2X:** Double the USART transmission speed – Tento bit má své využití pouze v případě asynchronního přenosu. Používáme-li synchronní přenos, nastavíme bit do log. 0. Nastavíme-li tento bit do log. 1, rychlost přenosu asynchronní komunikace se zdvojnásobí
- **Bit 0 – MPCM:** Multi-processor Communication Mode – Bit MPCM povoluje multiprocesorovou komunikaci. Je-li bit MPCM nastaven na log. 1 všechny příchozí rámce přijímány přes USART, které neobsahují informaci o adrese, se ignorují. Vysílač modulu USART nijak neovlivňuje nastavení bitu MPCM.

- Registr USCRB (Control and Status Register B)

Osmi bitový řídicí a stavový registr, určen pro nastavování USART , zjišťování stavu USART a nastavení přerušení.

Bit	7	6	5	4	3	2	1	0	
	<b>RXCIE</b>	<b>TXCIE</b>	<b>UDRIE</b>	<b>RXEN</b>	<b>TXEN</b>	<b>UCSZ2</b>	<b>RXB8</b>	<b>TXB8</b>	<b>USCRB</b>
čtení/zápis	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Počáteční hod.	0	0	0	0	0	0	0	0	

Obrázek 15: Registr USCRB (Control and Status Register B), zdroj[6]

Popis jednotlivých bitů v registru USCRB:

- **Bit 7 – RXCIE:** RX Complete Interrupt Enable – Nastavením tohoto bitu do log. 1 povolujeme přerušení od příznaku RXC. Přerušení bude generováno jenom v případě, že bit RXCIE je nastaven do log. 1, příznak globálního přerušení SREG je nastaven do log. 1 a bit RXC v registru UCSRA, je nastaven do log 1.
- **Bit 6 – TXCIE:** TX Complete Interrupt Enable - Nastavením tohoto bitu do log. 1 povolujeme přerušení od příznaku TXC. Přerušení bude generováno jenom v případě, že bit TXCIE je nastaven do log. 1, příznak globálního přerušení SREG je nastaven do log. 1 a bit TXC v registru UCSRA, je nastaven do log 1.
- **Bit 5 – UDRIE:** USART Data Register Empty Interrupt Enable - Nastavením tohoto bitu do log. 1 povolujeme přerušení od příznaku UDRE. Přerušení bude generováno jenom v případě, že bit UDRE je nastaven do log. 1, příznak globálního přerušení SREG je nastaven do log. 1 a bit UDRE v registru UCSRA, je nastaven do log 1.
- **Bit 4 – RXEN:** Receiver Enable – Je-li tento bit nastaven do log. 1, povolujeme přijímač modulu USART. Přijímač přenastaví normální režim portu na přijímací pin. Zakázáním přijímače se vymaže přijímací buffer a příznakové bity FE, DOR a PE se nastaví do log. 0
- **Bit 3 – TXEN:** Transmitter Enable - Je-li tento bit nastaven do log. 1, povolujeme vysílač modulu USART. Vysílač přenastaví normální režim portu na vysílací pin. Zakázání vysílače (nastavení bitu TXEN do log. 0) nabyde platnosti až v případě dokončení přenosu (vysílací posuvný registr a vysílací buffer neobsahují data k vyslání). Při zakázání tohoto bitu, nebude nadále port sloužit jako vysílací.
- **Bit 2 – UCSZ2:** Character Size – Tento bit spolu s bitem UCSZ1:0 v registru UCSRC nastavuje počet datových bitů (velikost znaku) v rámci.
- **Bit 1 – RXB8:** Receive Data Bit 8 – Je-li při přijímání znaků používáno devět bitů, je příznakový bit RXB8 právě devátý. Musí být přečten před čtením nižších bitů z registru UDR
- **Bit 0 – TXB8:** Transmit Data Bit 8 - Je-li při vysílání znaků používáno devět bitů je příznakový bit TXB8 právě devátý. Musí být zapsán před zapsáním nižších bitů do registru UDR

- Registr USCRC (Control and Status Register C)

Osmi bitový řídicí a stavový registr, určen pro nastavování USART a zjišťování stavu USART.

Tento registr sdílí stejné logické umístění jako UBRRH registr.

Bit	7	6	5	4	3	2	1	0	
	<b>URSEL</b>	<b>UMSEL</b>	<b>UPM1</b>	<b>UPM0</b>	<b>USBS</b>	<b>UCSZ1</b>	<b>UCSZ0</b>	<b>UCPOL</b>	<b>USCRC</b>
čtení/zápis	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Počáteční hod.	1	0	0	0	0	1	1	0	

Obrázek 16: Registr USCRC (Control and Status Register C), zdroj[6]

Popis jednotlivých bitů v registru USCRC:

- **Bit 7 – URSEL:** Register Select – Na základě tohoto bitu se rozhoduje, zda budeme přistupovat k registru USCRC nebo UBRRH. Při zápisu i při čtení z registru USCRC je tento bit nastavit na log. 1.
- **Bit 6 – UMSEL:** USART Mode Select – Tento bit vybírá mezi Asynchronním a synchronním režimem. Je-li v log.1, je nastaven režim synchronní, v log. 0 nastavíme USART do režimu asynchronního.
- **Bit 5:4 – UPM1:0:** Parity Mode – Tyto bity povolují a nastavují typ generování parity a kontrolu parity. Je-li povolen, vysílač bude automaticky generovat a posílat paritní bit v každém datovém rámcí. Příjímáče bude automaticky generovat paritu pro příchozí data a porovnávat ji s nastavením UPM0. Nastane-li neshoda, je nastaven příznakový bit PE v registru UCSRA.

UPM1	UPM0	Mód Parity
0	0	Parita zakázána
0	1	Parita rezervována
1	0	Povolena sudá parita
1	1	Povolena lichá parita

Tabulka 4: Nastavení UPM bitu, zdroj[6]



- **Bit 3 – USBS:** Stop Bit Select - Tento bit vybírá počet stop bitů, které budou vysílačem vloženy. Příjímač toto nastavení ignoruje. Nastavením tohoto příznakového bitu do log. 0, je přidán jeden stop bit. Je-li v log. 1, jsou vysílány stop bity dva.
- **Bit 2:1 – UCSZ1:0:** Character Size – Tyto bity v kombinaci s bitem UCSZ2 v registru UCSRB nastavují počet datových bitů (velikost znaku) používající v rámci vysílače a přijímače. V následující tabulce jsou uvedeny možné kombinace nastavení počtu bitů.

UCSZ2	UCSZ1	UCSZ0	Velikost znaku
0	0	0	5 bitů
0	0	1	6 bitů
0	1	0	7 bitů
0	1	1	8 bitů
1	0	0	Rezervováno
1	0	1	Rezervováno
1	1	0	Rezervováno
1	1	1	9 bitů

Tabulka 5: Nastavení velikosti znaku rámce, zdroj[6]

- **Bit 0 – UCPOL:** Clock Polarity – Tento bit je používán pouze pro synchronní režim. Při používání asynchronního režimu je tento bit nastaven do log. 0. Pomocí tohoto bitu se nastavuje vztah mezi výstupní změnou dat, vstupními vzorkovanými daty a synchronním hodinovým signálem (XCK).

UCPOL	Změna vysílaných dat (výstup na TxD pinu)	Přijímaná vzorkovaná data (vstup na RxD pinu)
0	Vzestupná XCK hrana	Sestupná XCK hrana
1	Sestupná XCK hrana	Vzestupná XCK hrana

Tabulka 6: Nastavení bitu UCPOL, zdroj[6]

- Registr UBRR (USART Baud Rate Registers)

Jedná se o 16 - ti bitový registr, rozdělen na dva 8 bitové celky UBRRH a UBRRL. Tento registr slouží k nastavení přenosové rychlosti. Registr UBRRH sdílí stejné logické umístění s registrem UCSRC.

Bit	15	14	13	12	11	10	9	8	
	URSEL	-	-	-	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
čtení/zápis	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Počáteční hod.	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Obrázek 17: Registr UBRR, zdroj[6]

#### Popis registru UBRR

- **Bit 15 – URSEL:** Register Select – Tento bit vybírá, zda se přistupuje k registru UBRRH nebo UCSRC. Při čtení z registru UBRRH i při zápisu do UBRRH musí být tento bit nastaven na log. 0.
- **Bit 14:12 – Reserved Bits** – Tyto bity jsou rezervovány pro budoucí použití. Pro kompatibilitu a pro zápis musí být tyto bity nastaveny na log. 0.
- **Bit 11:0 – UBRR11:0:** USART Baud Rate Register – Jedná se o 12- bitový registr obsahující údaje přenosové rychlosti jednotky USART. Registr UBRRH obsahuje 4 bity nejvyšší váhy, registr UBRRL obsahuje 8 bitů nižší váhy. Jestliže je změněná přenosová rychlost v průběhu přenosu dat, tyto data jsou následně poškozena. Zápisem do registru UBRRL se automaticky změní přenosová rychlost komunikace.

#### 4.2.5 Výpočet přenosové rychlosti USART

Pomocí vzorců uvedených v *tabulce 7*, lze vypočítat přenosovou rychlost v Baudech nebo obsah registru UBRR, za předpokladu, že máme danou přenosovou rychlost. Vypočítaná hodnota obsahu registru UBRR musí být vždy nejlépe celé číslo, v opačném případě není přenos dat přes USART bezchybný. Podrobnější popis této problematiky je popsán v katalogovém listu mikroprocesoru Atmel AVR ATmega8[5].

Přenosovou rychlost a obsah registru UBRR lze počítat pro tři režimy přenosu:

- Asynchronní normální režim
- Asynchronní režim se zdvojnásobenou rychlostí přenosu
- Synchronní ,Master‘ režim

Mód	Rovnice pro výpočet přenosové rychlosti	Rovnice pro výpočet hodnoty UBRR
Asynchronní Normální mód (U2X = 0)	$Baud = \frac{f_{osc}}{16(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16BAUD} - 1$
Asynchronní mód s dvojnásobnou rychlostí (U2X = 1)	$Baud = \frac{f_{osc}}{8(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8BAUD} - 1$
Synchronní mód	$Baud = \frac{f_{osc}}{2(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2BAUD} - 1$

Tabulka 7: Výpočet přenosové rychlosti, zdroj[6]

- BAUD – Přenosová rychlost (v bitech za sekundu)
- $f_{osc}$  - Frekvence oscilátoru
- UBRR - Obsah UBRRH a UBRRRL Registrů (0 - 4095)

#### 4.2.6 Komunikace po RS232

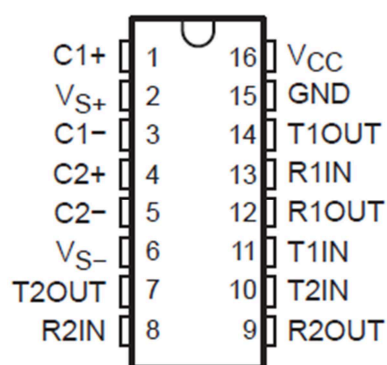
Jelikož napěťové úrovně rozhraní RS232 a standartu TTL jsou různé, nemůžeme přímo připojit rozhraní RS232 k rozhraní USART mikroprocesoru. V této podkapitole se tedy budeme zabývat problémem převodu napěťových úrovní rozhraní RS232 na úrovně napětí TTL. Rozhraní RS232 používá pro přenos dat dvě různá napětí, která jsou bipolární. Mohou nabývat napětí až +15V a -15V. Logická 0 bývá nejčastěji představována záporným napětím, zatímco logická 1 napětím kladným. Logika TTL používá mnohem menší napětí. Pohybuje se od 0V do 5V. K převodu napětí a zajištění kompatibility je tedy nutné použít převodník.

---

## 5 Převodník RS232

V této práci je pro převod z napěťové úrovně RS232 do úrovně TTL použit převodník MAX232IN.

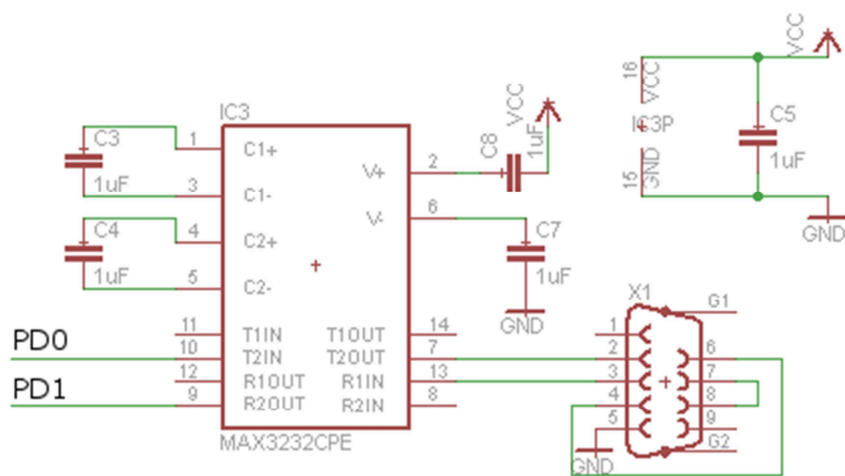
Převodník umožňuje přenos dat rychlosti až 120 kbit/s, napájen je napětím 5V. Obsahuje dva řídicí a vysílací obvody. Vstupní napětí se může pohybovat maximálně od -30V do 30V. Detailní popis toho obvodu lze nalézt v katalogovém listu výrobce [7].



Obrázek 18: Obvod MAX232IN a značení pinů, zdroj[7]

### 5.1 Praktické zapojení obvodu MAX232IN

Rozhraní RS232 je vyvedeno z počítače pomocí samce konektoru DB-9. V této práci je tedy tento konektor použit. Zapojení samice konektoru DB-9 je zobrazeno na *Obrázku 19*. Samotný obvod MAX232IN potřebuje pro svou funkčnost minimum součástek. Je použito 5 kondenzátorů o hodnotě 1uF. Vysílací pin T2OUT je připojen k pinu 2 konektoru DB-9. Příjímací pin R2IN je připojen k pinu 3 konektoru DB-9. Dále jsou vyvedeny piny T2IN, R2OUT do vstupů TxD a RxD USART linky mikroprocesoru. Napájení obvodu je zajištěno 5V. Na *Obrázku 19* je zobrazeno celé funkční zapojení obvodu MAX232IN.



Obrázek 19: Funkční zapojení Obvodu MAX232, zdroj: autor

---

## 6 LCD displej

Pro zobrazování výsledku měření slouží LCD displej MC16021E-SYR. Tento LCD modul obsahuje dobře známý řadič HD44780, který umožňuje snadnější práci s displejem (komunikační rozhraní, předdefinované znaky). Použitý LCD displej je dvouřádkový, přičemž na každý řádek je možné zobrazit až 16 znaků. Z modulu je vyvedeno 16 pinů, piny 15 a 16 jsou volitelné a slouží k napájení podsvětlení displeje. V této práci použitý displej nedisponuje touto možností, a proto nejsou piny 15 a 16 zapojeny.

Každý jednotlivý znak na LCD displeji má svou pevně danou adresu. Přes tuto danou adresu lze vypisovat znaky na přesně určená místa displeje. Na *Obrázku 20* jsou uvedeny adresy a k nim příslušné pozice na LCD displeji.

Znak	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Řádek 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Řádek 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

*Obrázek 20: Adresy znaku LCD displeje, zdroj[8]*

Číslo pinu	Popis pinu
1	Vss - GND
2	Vdd – Napájecí napětí (3,3V až 5V)
3	Vo – Nastavení kontrastu displeje přes potenciometr
4	RS – Register Select – určuje, zda se přenáší data (log. 1) nebo instrukce (log. 0)
5	R/W – Tento pin nastavuje, zda se čte nebo na displej zapisuje Obvykle bývá nastaven na log. 0
6	E – enable; zapíná nebo vypíná displej
7	DB0 – datový bit
8	DB1 – datový bit
9	DB2 – datový bit

---

10	DB3 – datový bit
11	DB4 – datový bit
12	DB5 – datový bit
13	DB6 – datový bit
14	DB7 – datový bit
15	A – Anoda (+), podsvícení
16	K – Katoda (-), podsvícení

*Tabulka 8: Popis pinů, zdroj[8]*

LCD modul pracuje ve čtyř bitovém nebo osmi bitovém režimu. Jedním z důvodů pro výběr 4 bitového režimu může být například nedostatek volných pinů mikroprocesoru. U tohoto režimu se posílají jako první horní 4 bity, následně dolní 4 bity. K přenosu datových bitů jsou použity piny DB4 až DB7. U 8 bitové komunikace je posíláno 8 bitů najednou za pomoci pinů DB0 až DB7. V naší práci budeme používat komunikaci pomocí 4 datových vodičů a 3 pinů řídicích.

Pro snadnější manipulaci a práci s LCD displejem s řadičem HD44780 jsou již vytvořeny knihovny, které obsahují funkce pro řízení modulu. Mezi známé knihovny patří knihovna Petra Fleryho [9] nebo alank2 knihovna [10]. V naší práci se seznámíme s knihovnou alank2 a s její pomocí budeme komunikovat s LCD displejem.

## 6.1 Popis knihovny alank2

V této kapitole si popíšeme knihovnu alank2 a metody použité v této bakalářské práci. Knihovna alank2 obsahuje dva hlavičkové soubory. Hlavičkový soubor hd44780.h definuje konstanty a metody používané pro práci s displejem. Druhý hlavičkový soubor hd44780\_settings.h obsahuje nastavení komunikace a příslušných portů, přes které bude komunikace probíhat.

### 6.1.1 Hlavičkový soubor hd44780.h

Použité funkce:

- void lcd\_init() – inicializace lcd displeje (nastavení módu, zapnutí displeje)
- void lcd\_clrscr() – vymazání displeje
- void lcd\_puts(const char \*s) – výpis textu na displej
- void lcd\_goto(uint8\_t pos) – nastavení pozice na displeji

---

### 6.1.2 Hlavičkový soubor `hd44780_settings.h`

Použité konstanty:

- `F_CPU` – nastavení frekvence mikroprocesoru, nastaveno na 1000000
- `LCD_BITS` - nastavení režimu displeje, v našem případě je tato konstanta nastavena na číslo 4, pokud by byl použitý osmi bitový režim, byla by nastavena na číslo 8

Jelikož používáme 4 bitový režim, je potřeba nastavit porty a piny mikroprocesoru u příslušných pinů LCD displeje (DB4-DB7) přes které budeme komunikovat, včetně RS pinu a pinu enable. Pin R/W nebude použit.

- `LCD_DB4_PORT` - nastaven `PORTC`
- `LCD_DB4_PIN` – nastaven pin 0
- `LCD_DB5_PORT` - nastaven `PORTC`
- `LCD_DB5_PIN` – nastaven pin 1
- `LCD_DB6_PORT`- nastaven `PORTC`
- `LCD_DB6_PIN` – nastaven pin 2
- `LCD_DB7_PORT` – nastaven `PORTC`
- `LCD_DB7_PIN` - nastaven pin 3
- `LCD_RS_PORT` - nastaven `PORTC`
- `LCD_RS_PIN` – nastaven pin 4
- `LCD_E_PORT` - nastaven `PORTC`
- `LCD_E_PIN` – nastaven pin 5
- `RW_LINE_IMPLEMENTED` 0

Jelikož knihovna `alank2` podporuje připojení až čtyř displejů, je třeba nastavit počet pouze jednoho displeje:

- `LCD_DISPLAYS` 1

Dále je potřeba nastavit počet řádků displeje na dva řádky:

- `LCD_DISPLAY_LINES` 2



---

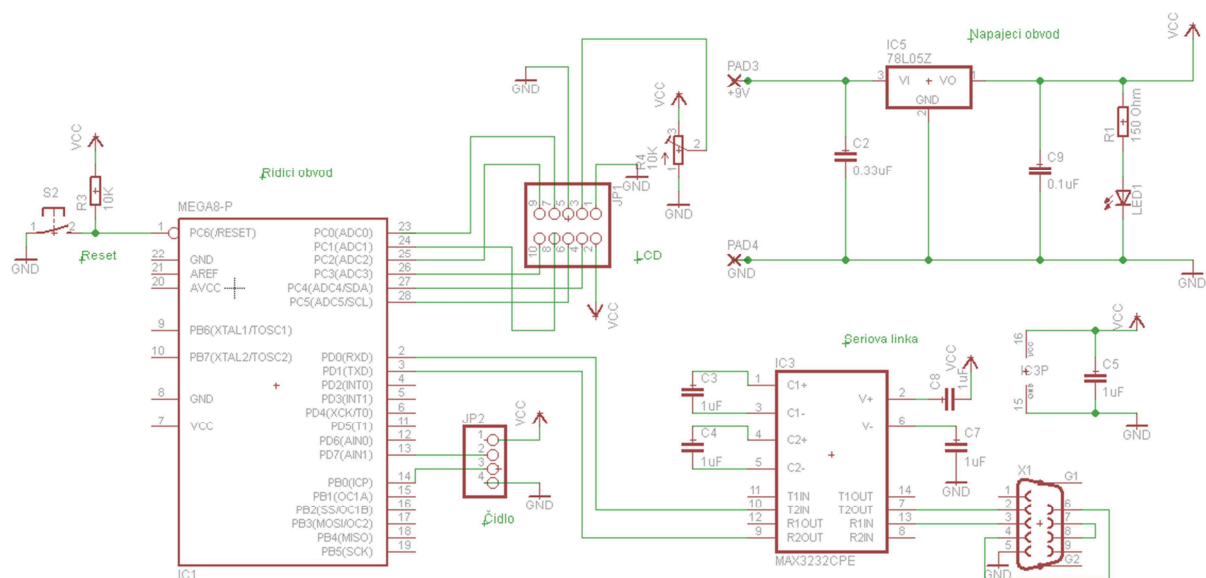
## 7 Návrh desky plošného spoje

Návrh desky plošného spoje (dále jen DPS) byl realizován v programu Eagle [11]. Tento software slouží k vytvoření obvodu, kreslení schémat a návrhu DPS [12].

### 7.1 Schéma zapojení

Schéma zapojení můžeme rozdělit do 3 logických bloků. První blok realizuje napájecí obvod, druhý blok je určen pro přenos dat přes sériovou linku, a třetí blok tvoří řídicí obvod, včetně LCD displeje a měřicího modulu. K sestavení celého obvodu realizujícího digitální měřidlo hladiny kapaliny jsou nezbytné následující součástky a komponenty:

- Napájecí obvod [13]
  - Napěťový stabilizátor L7805 – určen pro stabilizování napětí na 5V
  - Kondenzátory – 0,33  $\mu$ F a 0,1  $\mu$ F
  - Odpor 150 Ohm
  - Zelená LED dioda
- Sériová linka
  - Integrovaný obvod MAX232IN
  - Kondenzátory – 5x 1 $\mu$ F
  - Konektor DE-9 F [14]
- Řídicí obvod
  - Mikroprocesor ATmega 8
  - Odpor 10K Ohm
  - Tlačítko pro funkci resetu mikroprocesoru
  - Trimr 10K Ohm
  - LCD Displej (10 pinový konektor)
  - Měřicí modul (4 pinový konektor)



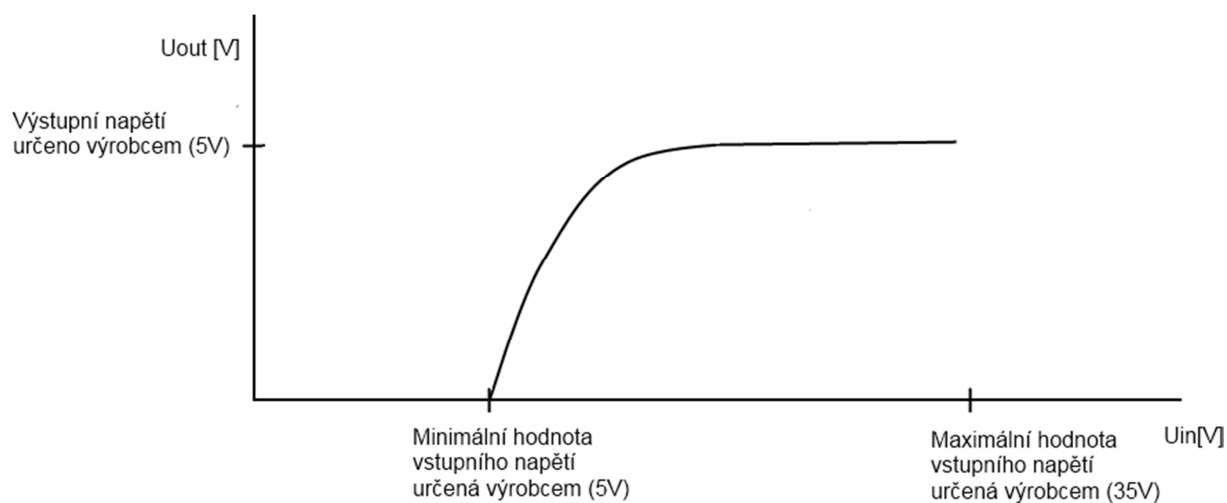
Obrázek 21: Schéma zapojení, zdroj: autor

## 7.2 Popis napájecího obvodu

V této kapitole se blíže seznámíme s obvodem pro napájení zařízení. Obvod pro sériovou linku a řídicí obvod již byl popsán v předchozích kapitolách.

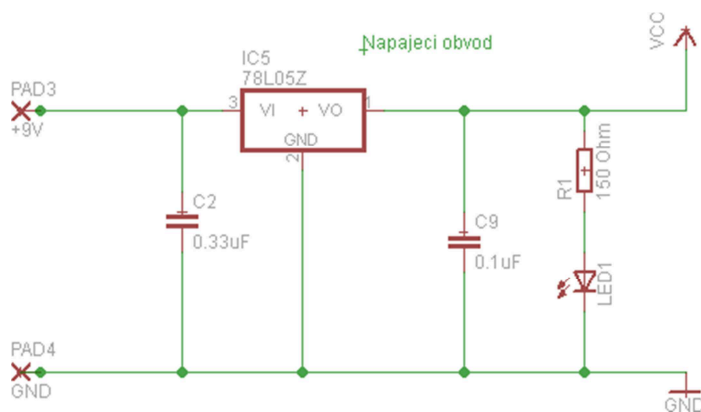
Hlavní článkem napájecího obvodu je integrovaný stabilizátor L7805. Stabilizátor se vyznačuje tím, že stabilizuje výstupní hodnotu napětí na stálou hodnotu, změní-li se hodnota vstupního napětí nebo změní-li se zatěžovací proud.

Z napěťové charakteristiky stabilizátoru (obrázek 22) můžeme vidět, že při přivedení minimálního napětí (v našem případě 5 V) se na výstupu objeví stabilizované napětí (5V) a zůstane konstantní i při dalším zvyšování vstupního napětí a to až do jeho maximální meze, a to 35V.



Obrázek 22: Napěťová charakteristika stabilizátoru, zdroj [15]

Napájecí obvod tedy tvoří stabilizátor L7805, dva blokovací kondenzátory, jejichž kapacita je dána výrobcem stabilizátoru a LED dioda s ochranným odporem, pro indikování funkčnosti.



Obrázek 23: Napájecí obvod, zdroj: autor

Primárním účelem blokovacích kondenzátorů, které se v obvodu nacházejí, je absorpce rušivých signálů vyšší frekvence ve stejnosměrném průběhu napětí. Fungují taktéž jako krátkodobé zdroje energie, které se při rychlé změně zatěžovacího proudu částečně vybíjejí a vyrovnávají tuto rychlou změnu, než na ní zareaguje zdroj napětí.

Předřadný odpor zelené LED diody je vypočítán podle následujícího vzorce:

$$R = \frac{(U_{\text{zdroje}} - U_{\text{LED}})}{I_{\text{LED}}} [\Omega]$$

$U_{\text{zdroje}}$  – Napájecí napětí

$U_{\text{LED}}$  – Napětí na LED diodě

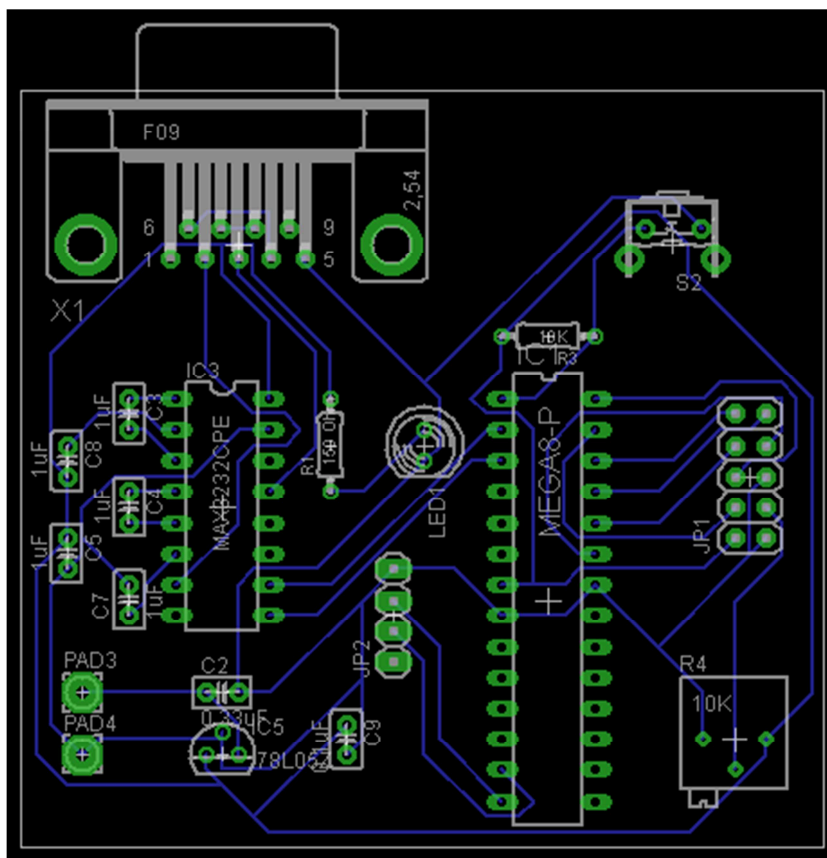
$I_{\text{LED}}$  – Maximální protékající proud procházející diodou

Po dosazení hodnot do vzorce pro námi použitou diodu ( $U_{LED} = 2,2 \text{ V}$ ,  $I_{LED} = 0,020 \text{ A}$  [16]), dostaneme hodnotu odporu  $140 \Omega$ , použijeme tedy odpor, který se nejbližší rovná této hodnotě, a to  $150 \Omega$ .

$$R = \frac{(5 - 2,2)}{0,020} = 140 \approx 150 \Omega$$

### 7.3 DPS

Po nakreslení a vyhotovení schématu zapojení, bylo možné vytvořit návrh desky plošného spoje v programu Eagle. Deska je osazena všemi výše popsanými součástkami a její rozměry jsou  $66 \times 62 \text{ mm}$ .



Obrázek 24: DPS, zdroj: autor

---

## 8 Firmware

Základní požadavek na vzniklý firmware, bylo změření výšky hladiny kapaliny v nádrži a následně vypočet objemu této kapaliny, za předpokladu znalosti rozměru nádrže. Další požadavkem, bylo výsledný objem kapaliny zobrazit na LCD displeji a přes sériové rozhraní RS232 odesílat výsledky měření, například do PC. Program zajišťující tuto funkčnost je realizován pomocí hlavního programu, výše zmiňovanými 4 funkcemi pro práci s LCD displejem, a 4 funkcemi, popsány níže. V následujících kapitolách si popíšeme použité funkce a důležité části použitých kódů. Kompletní program i s popisujícími komentáři je obsažen ve zdrojovém kódu firmwaru. Výše zmiňované 4 funkce se nazývají následovně:

- `void moduleInit()` – Funkce inicializuje ultrazvukové čidlo
- `void uart_initUsart(uint16_t baud)` – funkce inicializuje USART
- `void uart_puts(const char *s)` – vypsání znaku v hyperterminálu
- `void uart_gets( char* input, int maxlen )` – přijetí řetězce přes sériovou linku a následně jeho uložení

### 8.1 Popis dílčích funkcí

V této kapitole se budeme zabývat popisem celého programu a jeho části. Popis se bude zabývat použitými funkcemi a hlavním programem.

- `void moduleInit ()`

Tato funkce inicializuje měřicí ultrazvukové čidlo. Inicializace se provádí za pomoci 10 mikrosekundového impulsu vyslaného na pin TRIG čidla.

```
PORTD &= ~(1 << PD7);  
_delay_us(2);  
PORTD |= (1 << PD7);  
_delay_us(10);  
PORTD &= ~(1 << PD7);
```

---

Po inicializaci čidla mikroprocesor naslouchá na pinu PB0(ICP1), a čeká na příchod ECHO pulsu z ultrazvukového čidla. Je-li na pinu PB0 zaregistrována změna stavu z log. 0 do log. 1, nastane přerušení a toto přerušení je obslouženo. Pokud je příznakový bit ICES1 v registru TCCR1B v log. 1, je obsah čítače uložen do proměnné OldValue a příznakový bit ICES1 je nastaven do log. 0, což zajistí přerušení, nastane-li změna signálu na pinu PB0 z log. 1 do log. 0. Je-li tato změna mikroprocesorem zaregistrována, vypočítá se doba trvání ECHO pulsu a bit ICES1 je nastaven zpět do log. 1, pro další měření.

```
ISR(TIMER1_CAPT_vect)
{
    static uint16_t NewValue = ICR1;
    static uint16_t OldValue;

    if( !(TCCR1B & (1<<ICES1)) )

    {
        PulseDuration = NewValue - OldValue;
        TCNT1 = 0;
    }

    OldValue = NewValue;
    TCCR1B ^= (1<<ICES1);
}
```

- void initUsart (uint16\_t baud)

Tato funkce nastaví USART. Je nastaven asynchronní mód s jedním stop bitem a osmi datovými bity.

Na základě přenosové rychlosti je do proměnné ubrr uložen vypočítaný obsah registru UBRR

```
uint16_t ubrr = 1000000/16/ baud-1;
```

Poté je UBRR registr nastaven

---

```
UBRRH = (unsigned char) (ubrr>>8);
UBRRL = (unsigned char) ubrr;
```

V další části je povolena činnost vysílače, přijímače a je povoleno přerušení.

```
UCSRB = (1<<RXEN) | (1<<TXEN) | (1<<RXCIE);
```

Následuje nastavení formátu rámce v registru UCSRC. Délka znaků 8 bitů, 1 stop bit. Nastavení se provede vybráním registru USRCR, zapsáním log. 1 do příznakových bitů URSEL, UCSZ1 a UCSZ2. Pro nastavení jednoho stop bitu zůstává příznakový bit USBS v log. 0.

```
UCSRC=(1<<URSEL) | (1<<UCSZ1) | (1<<UCSZ0);
```

- void uart\_puts(const char \*s )

Funkce vyšle celý řetězec. Vysílá znak po znaku, dokud nedosáhne posledního znaku. Tato funkce je použita z knihovny uart.h od Petra Fleuryho.

```
while (*s)
    uart_putc(*s++);
```

- void uart\_gets( char\* input, int maxlen )

Cílem této funkce je odesíláním dat (textových řetězců) zadaných klávesnicí do hyperterminálu, přes sériovou linku do PC. Funkce je použita z knihovny uart.h od Petra Fleuryho a mírně modifikována. Funkce vypisuje zadávané znaky na obrazovku a poté je ukládá do pole znaků. Přenos končí v případě, je-li zadán znak konce řádku (zmačknutí klávesy enter).

```
uart_putc( c );
if (c != '\r' && strlen <= maxlen-1)
{
    strlen++;
    *input = c;
    input++;
}
```

---

```
    }  
    else  
    {  
        *input='\0';  
  
        return;  
    }
```

## 8.2 Popis hlavního programu

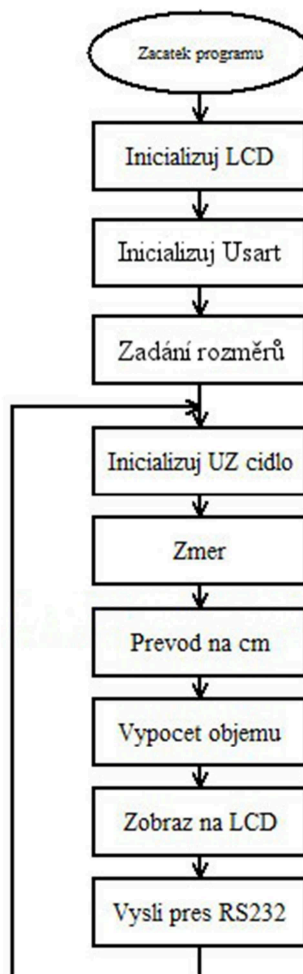
Hlavní program tvoří mimo jiné nekonečná smyčka, která zajišťuje kontinuální měření, zobrazování výsledku LCD displejem a odesíláním hodnot přes RS232 rozhraní. Zdrojový kód nekonečné smyčky hlavního programu je tvořen již výše popsány funkcemi a dalšími doplňujícími řádky kódu. Na celý kompletní program je možné se podívat ve zdrojovém kódu firmwaru. Měření probíhá v následujících krocích:

- Zadání tvaru podstavy nádrže
- Nastavení rozměrů nádrže
- Inicializace ultrazvukového čidla
- Změření vzdálenosti
- Převedením naměřených hodnot na centimetry
- Výpočet objemu nádrže
- Zobrazení hodnot na LCD displeji
- Vyslání hodnot (objemu) přes sériový port

Jelikož ultrazvukové čidlo měří hodnoty až od dvou centimetrů, je program upraven tak, že je potřeba čidlo umístit dva centimetry nad maximální hodnotu hladiny měřené kapaliny. Na druhé straně dolní hladina, čili prázdná nádrž, musí být od čidla nanejvýš 498 cm.

Přes sériový port jsou odesílány pouze měnící se údaje, to znamená, že je-li stále měřená stejná hodnota, je odeslána pouze jednou.





Obrázek 26: Vývojový diagram hlavního programu, zdroj: autor

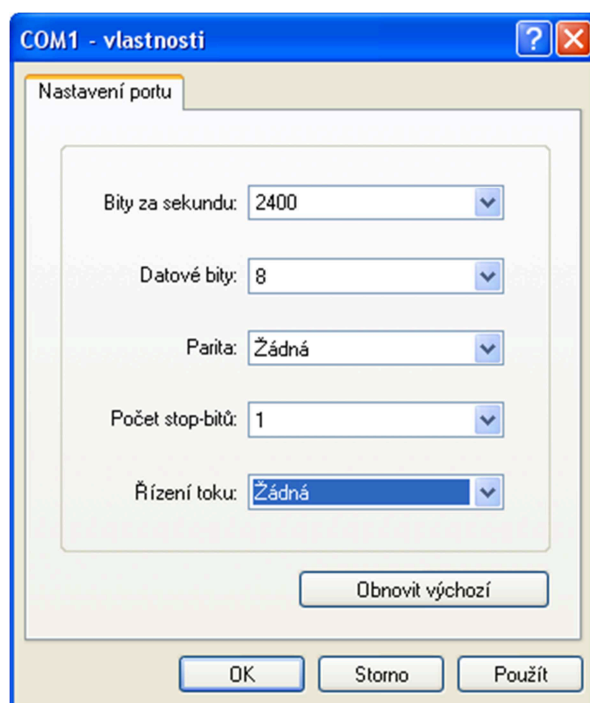
---

## 9 Testování zařízení

Pro fungování, základní inicializaci a tedy i testování zařízení je zapotřebí připojit toto zařízení přes sériový port například do PC. V PC spustíme nástroj zvaný hyperterminál, který je běžně dostupný ve WinXP, případně k obdobnému programu v jiných operačních systémech.

### 9.1 Nastavení sériového přenosu

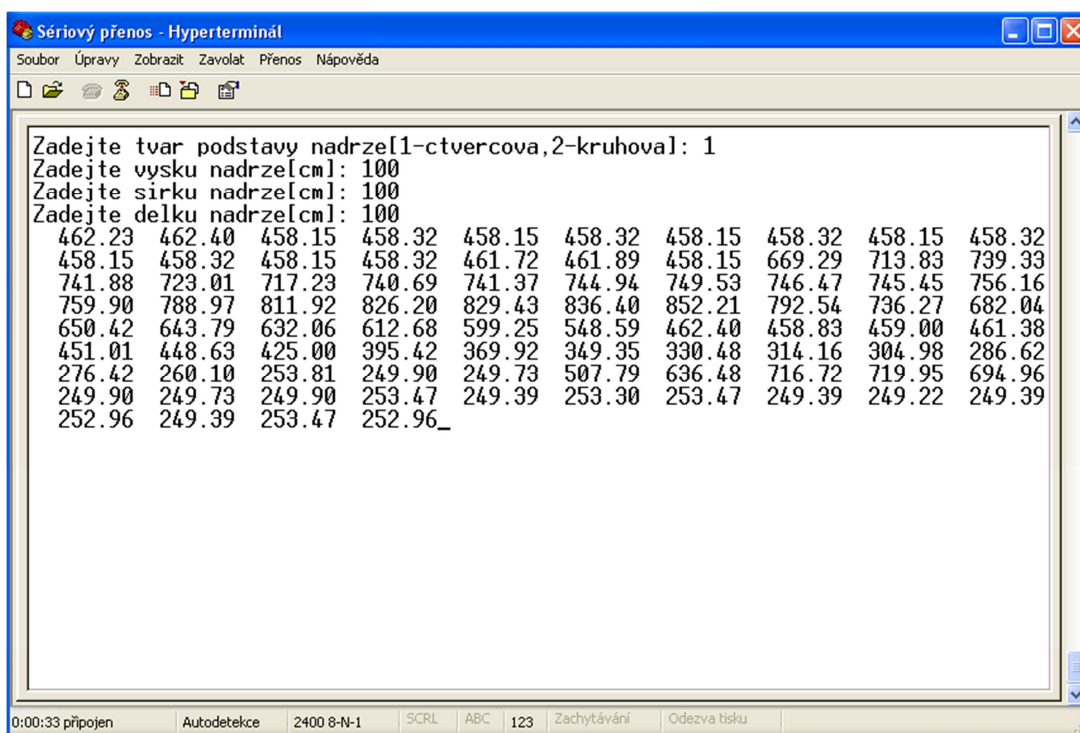
Aby došlo k bezchybnému přenosu dat, je potřeba nastavit příslušné parametry přenosu. V našem případě je potřeba nastavit přenosovou rychlost na 2400 baudů, 8 datových bitů, 1 stop bit, přenos bez kontroly paritou a bez řízení toku (Obrázek 27).



Obrázek 27: Nastavení parametrů přenosu, zdroj: autor

### 9.2 Inicializace měření

Před spuštěním měření je zapotřebí nastavit rozměry nádrže a její tvar podstavy. Tyto údaje se nastavují již přes sériovou linku připojenou od zařízení k PC. Po vložení parametru začne měření a v hyperterminálu jsou vypisovány výsledky měření (Obrázek 28).



Obrázek 28: Nastavení rozměrů nádrže a přenos dat, zdroj: autor

### 9.3 Přesnost zařízení

Zařízení bylo testováno na mnoha měřeních a bylo zkonstruováno a vyladěno co možná k největší přesnosti. Pro názornost je v *tabulce 9* uveden výsledek deseti měření zařízení. Je vidno, že chyba zařízení byla v řádech milimetrů. Největší odchylka byla 3 mm, nejmenší 0 mm. Chyba mohla být způsobena bezkontaktním ultrazvukovým čidlem nebo také nepřesným vnitřním oscilátorem mikroprocesoru ATmega8.

---

Měření číslo:	Naměřená hodnota [cm]	Skutečná hodnota [cm]
1	52,3	52,2
2	70,2	70,1
3	92,7	93
4	45,1	44,9
5	28,3	28
6	37,7	37,6
7	1,6	1,6
8	18,2	18
9	89,8	90
10	95,7	96

*Tabulka 9: Naměřené hodnoty, zdroj: autor*

---

## 10 Závěr

Tato bakalářská práce demonstrovala využití ultrazvukového bezkontaktního čidla Ultrasonic HC-SR04 připojeného k mikroprocesoru ATmega8, pro získání informace o výšce hladiny kapaliny v nádrži. Následně realizovala přepočty na celkový objem této kapaliny v nádrži, zobrazení výsledku měření na LCD displeji a vyslání naměřených dat přes sériovou linku.

Ze zadání je patrné, že bylo potřeba se blíže seznámit s mikroprocesorem ATmega8, jeho vlastnostmi, přednostmi a funkcemi, které by byly využity při tvorbě jak hardwarové části, tak i firmwaru. Totéž platí i pro ultrazvukové čidlo Ultrasonic HC-SR04, obvod umožňující komunikaci přes sériovou linku RS232 a taktéž pro LCD displej MC16021E-SYR. Dalo by se tedy říci, že bylo pro mě velkým přínosem pracovat se všemi výše zmiňovanými komponenty. Hlavně s jádrem celé aplikace, mikroprocesorem ATmega8, který je podle mého názoru pro využití v této aplikaci velmi dobrým řešením a to i z důvodu ISP programování. Postupným „přiváděním k životu“ těchto komponentů docházelo k vývoji zařízení a následně k jeho ladění. K vývoji firmwaru byl použit jazyk C, se kterým jsem se už v minulosti setkal při studiu na VŠB-TU, naopak s vývojem a návrhem desky plošných spojů v programu Eagle, jsem neměl žádné zkušenosti. Realizace této bakalářské práce mě tedy obohatila i v tomto směru.

Z pohledu dalšího vývoje tohoto zařízení, existují další možnosti, jak zdokonalit toto měřidlo. Příkladem je třeba použití novějšího rozhraní USB pro přenos dat do PC, doplnění zařízení o klávesnici, přes kterou by bylo možné měřidlo konfigurovat nebo připojení více měřících čidel k jednomu mikroprocesoru.

---

## 11 Použitá literatura

- [1] Ultrazvukové měření výšky hladiny kapalin a pevných látek. *Odbornecasopisy* [online]. 2001 [cit. 2013-03-26]. Dostupné z: [http://www.odbornecasopisy.cz/index.php?id\\_document=33552](http://www.odbornecasopisy.cz/index.php?id_document=33552)
- [2] Ultrazvukové vyšetření. *projekt-endoskopie* [online]. Neznámý [cit. 2013-03-14]. Dostupné z: [http://www.projekt-endoskopie.cz/attachment/ultrazvukove\\_vysetreni.doc](http://www.projekt-endoskopie.cz/attachment/ultrazvukove_vysetreni.doc)
- [3] Hustota. *Converter* [online]. 2001 [cit. 2013-03-12]. Dostupné z: <http://www.converter.cz/tabulky/hustota-pevne.htm>
- [4] Ultrasonic ranging module. *microsolutions*. [online]. 2010 [cit. 2013-03-12]. Dostupné z: <http://http://www.micropik.com/PDF/HCSR04.pdf>
- [5] AC - PG - USBASP USBASP AVR Programmer. *Protostack* [online]. 2012 [cit. 2013-03-12]. Dostupné z: <http://www.protostack.com/download/Users%20Guide%20%28AC-PG-USBASP-UG-V2.0%29.pdf>
- [6] ATmega8. *Atmel* [online]. 2013 [cit. 2013-03-12]. Dostupné z: [http://www.atmel.com/Images/Atmel-2486-8-bit-AVR-microcontroller-ATmega8\\_L\\_datasheet.pdf](http://www.atmel.com/Images/Atmel-2486-8-bit-AVR-microcontroller-ATmega8_L_datasheet.pdf)
- [7] Max232. *Ti* [online]. 2004 [cit. 2013-03-12]. Dostupné z: <http://www.ti.com/lit/ds/symlink/max232.pdf>
- [8] MC1602E. *Gme* [online]. 2009 [cit. 2013-03-12]. Dostupné z: <http://www.gme.cz/dokumentace/513/513-069/dsh.513-069.1.pdf>
- [9] *Peter Fleury* [online]. 2012 [cit. 2013-03-12]. Dostupné z: <http://homepage.hispeed.ch/peterfleury/index.html>
- [10] Hd44780 Character LCD Displays – Part 2. *Prostack* [online]. 2010 [cit. 2013-03-12]. Dostupné z: <http://www.protostack.com/blog/2010/04/character-lcd-displays-part-2/>
- [11] Návod na výrobu plošných spojů fotocestou. *Georgeb* [online]. 2009 [cit. 2013-03-12]. Dostupné z: <http://www.georgeb.wz.cz/dps.htm>
- [12] Plošný spoj. *Wikipedia* [online]. 2013 [cit. 2013-03-12]. Dostupné z: [http://cs.wikipedia.org/wiki/Plo%C5%A1n%C3%BD\\_spoj](http://cs.wikipedia.org/wiki/Plo%C5%A1n%C3%BD_spoj)

- 
- [13] Atmega8 breadboard circuit – Part 1 of 3 – Power suply. *Prostack* [online]. 2009 [cit. 2013-03-12]. Dostupné z: <http://www.protostack.com/blog/2009/07/atmega8-breadboard-circuit-part-1-of-3-power-supply/>
- [14] D-Sub. *Wikipedia* [online]. 2012 [cit. 2013-03-12]. Dostupné z: <http://cs.wikipedia.org/wiki/D-Sub>
- [15] Stabilizace pomocí obvodu 7805. *Elektroportal* [online]. 2013 [cit. 2013-04-26]. Dostupné z: <http://www.elektroportal.xf.cz/index.php?p=stabilizace-pomoci-obvodu-7805>
- [16] L-934SGD. *GME* [online]. 2013 [cit. 2013-04-26]. Dostupné z: <http://www.gme.cz/led-kulate-pouzdro/l-934sgd-p511-772/#popis>

---

## 12 Seznam obrázků

Obrázek 1:	Měřicí modul Ultrasonic HC-SR04, zdroj [4].....	3
Obrázek 2:	Rozměry modulu Ultrasonic HC-SR04, zdroj [4] .....	3
Obrázek 3:	Efektivní uhel pro měření, zdroj [4] .....	4
Obrázek 4:	Časový diagram komunikace s modulem, zdroj [4] .....	5
Obrázek 5:	Programátor, zdroj[5].....	6
Obrázek 6:	ISP 10 pinový konektor, zdroj[5] .....	7
Obrázek 7:	Rozložení pinů mikroprocesorů ATmega8, zdroj[6] .....	10
Obrázek 8:	Architektura čítače/časovače1, zdroj[6] .....	11
Obrázek 9:	Registr TCCR1B, zdroj[6].....	12
Obrázek 10:	Registr TIMSK, zdroj[6].....	13
Obrázek 11:	Datový rámec asynchronního přenosu, zdroj[6].....	14
Obrázek 12:	Blokové schéma USART, zdroj[6] .....	15
Obrázek 13:	Registr UDR, zdroj[6] .....	16
Obrázek 14:	Registr USCRA, zdroj[6].....	16
Obrázek 15:	Registr USCRB (Control and Status Register B), zdroj[6] .....	18
Obrázek 16:	Registr USCRC (Control and Status Register C), zdroj[6].....	19
Obrázek 17:	Registr UBRR , zdroj[6] .....	21
Obrázek 18:	Obvod MAX232IN a značení pinů, zdroj[7].....	23
Obrázek 19:	Funkční zapojení Obvodu MAX232, zdroj: autor .....	24
Obrázek 20:	Adresy znaku LCD displeje, zdroj[8] .....	25
Obrázek 21:	Schéma zapojení, zdroj: autor.....	29
Obrázek 22:	Napěťová charakteristika stabilizátoru, zdroj [15].....	30
Obrázek 23:	Napájecí obvod, zdroj: autor.....	30
Obrázek 24:	DPS, zdroj: autor .....	31
Obrázek 26:	Vývojový diagram hlavního programu, zdroj: autor.....	36
Obrázek 27:	Nastavení parametrů přenosu, zdroj: autor .....	37
Obrázek 28:	Nastavení rozměrů nádrže a přenos dat, zdroj: autor.....	38



---

## 13 Seznam tabulek

<i>Tabulka 1:</i>	<i>Hustota vody a vzduchu o běžném tlaku a teplotě 20°C, zdroj: [3]</i>	<i>2</i>
<i>Tabulka 2:</i>	<i>Tabulka podporovaných mikrokontrolérů, zdroj[5]</i>	<i>7</i>
<i>Tabulka 3:</i>	<i>Nastavení zdroje hodin a předděličky, zdroj[6]</i>	<i>12</i>
<i>Tabulka 4:</i>	<i>Nastavení UPM bitu, zdroj[6]</i>	<i>19</i>
<i>Tabulka 5:</i>	<i>Nastavení velikosti znaku rámce, zdroj[6]</i>	<i>20</i>
<i>Tabulka 6:</i>	<i>Nastavení bitu UCPOL, zdroj[6]</i>	<i>20</i>
<i>Tabulka 7:</i>	<i>Výpočet přenosové rychlosti, zdroj[6]</i>	<i>22</i>
<i>Tabulka 8:</i>	<i>Popis pinů, zdroj[8]</i>	<i>26</i>
<i>Tabulka 9:</i>	<i>Naměřené hodnoty, zdroj: autor</i>	<i>39</i>

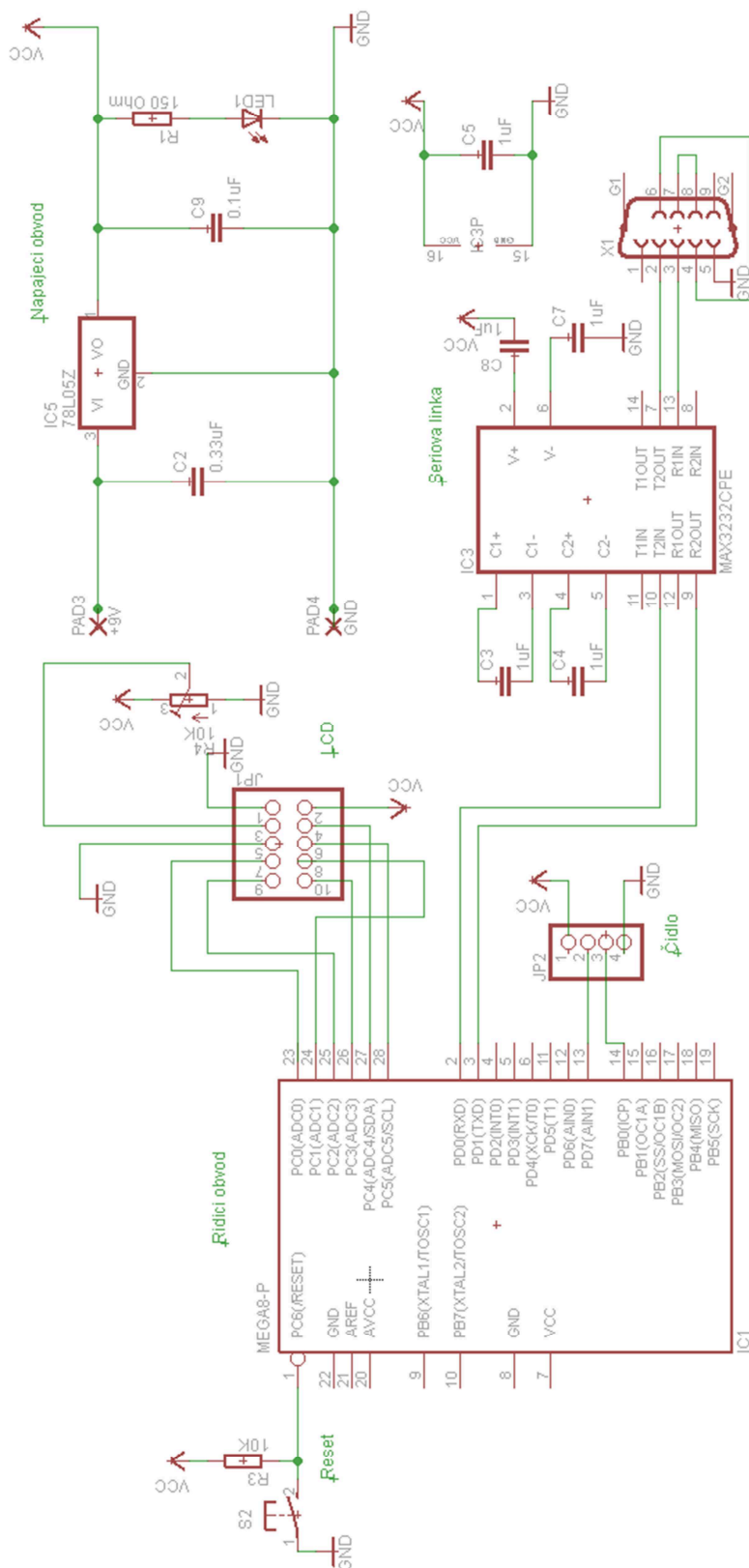
---

## 14 Seznam příloh

<b>Příloha I</b>	Schéma zapojení
<b>Příloha II</b>	DPS bez součástek
<b>Příloha III</b>	LCD modul

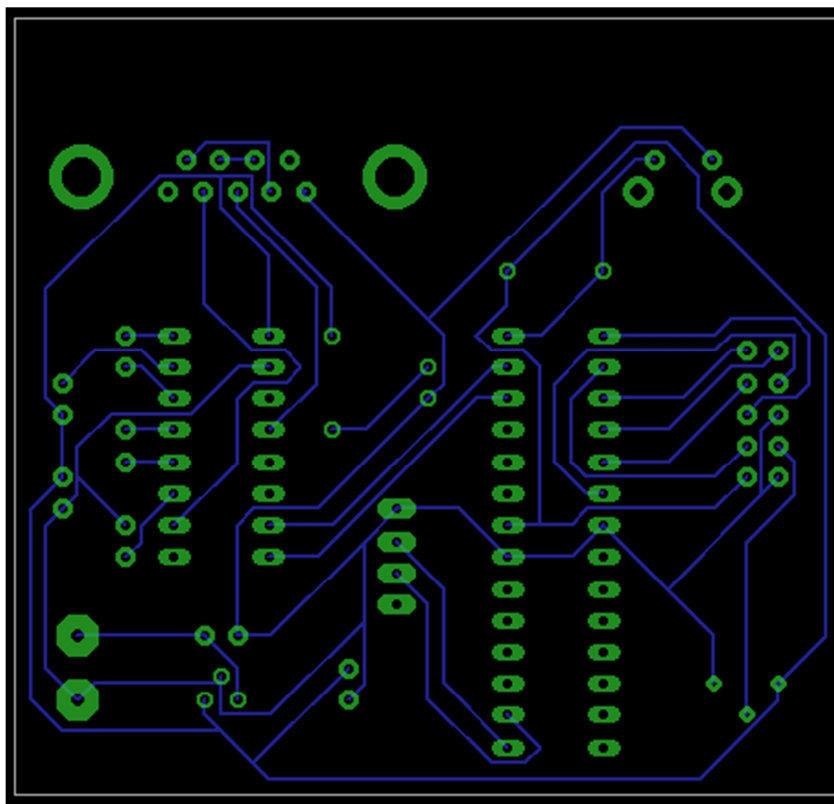
## Příloha I

## Schéma zapojení



---

**Příloha II**      DPS bez součástek



**Příloha III**    LCD modul

